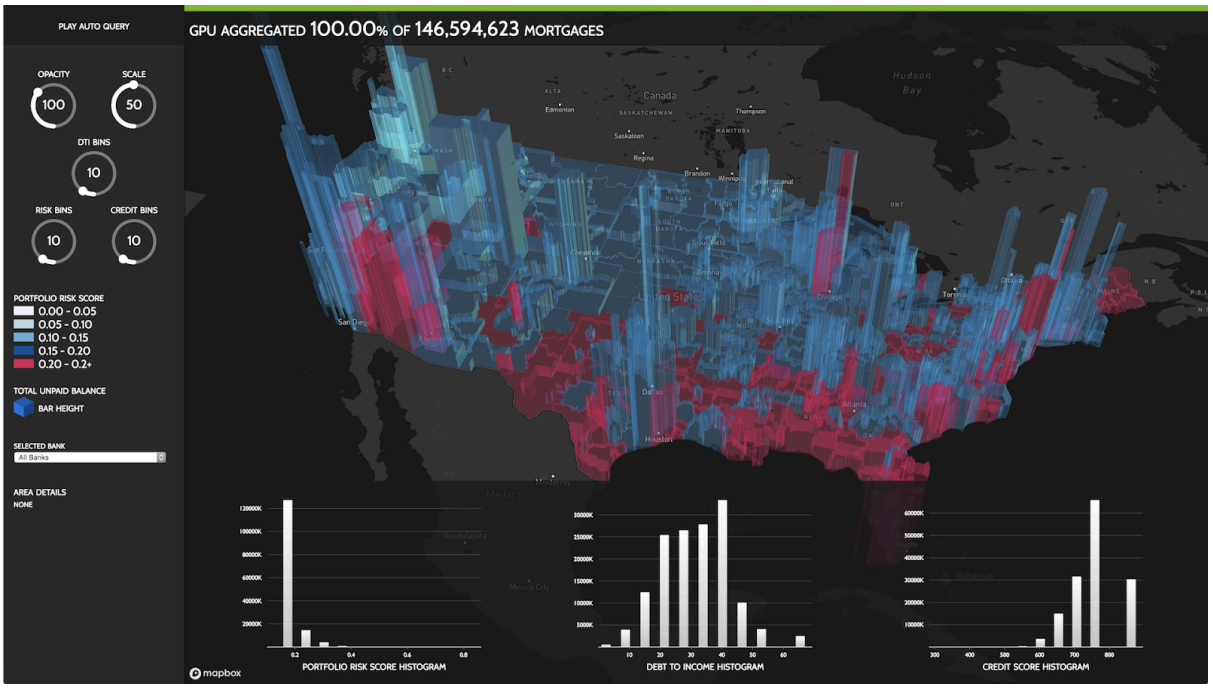
RAPIDS acelera la ciencia de datos de extremo a extremo

15 de octubre de 2018

Por Shashank Prasanna y Mark Harris



Los problemas actuales de la ciencia de datos exigen un aumento drástico de la escala de los datos, así como de la potencia computacional necesaria para procesarlos. Lamentablemente, el fin de la ley de Moore implica que el manejo de grandes cantidades de datos en el ecosistema de la ciencia de datos actual requiere escalar a muchos nodos de CPU, lo que trae sus propios problemas de cuellos de botella en las comunicaciones, consumo de energía y costos (ver figura 1).

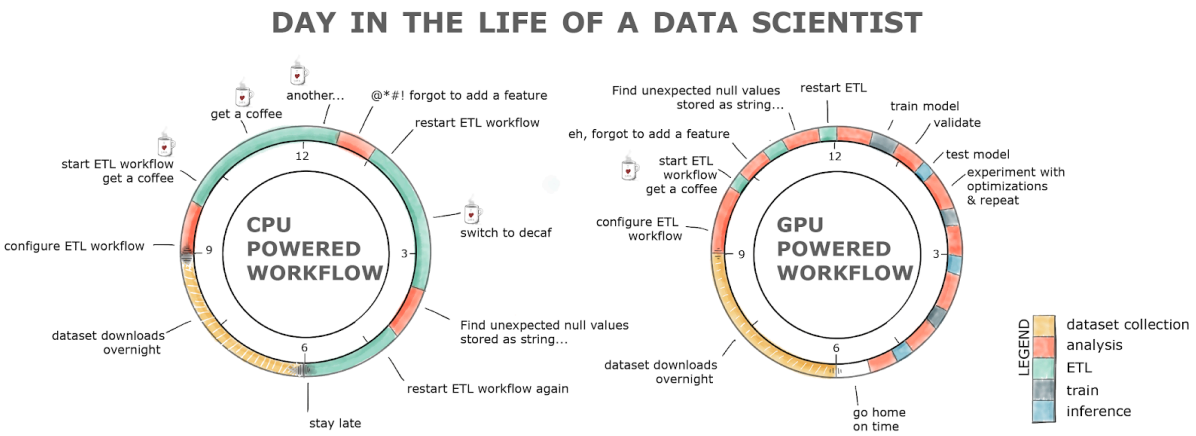


Figura 1. La ciencia de datos impulsada por GPU permite una interactividad mucho mayor, pero ofrece menos pausas para tomar café.

Una parte clave de la ciencia de datos es la exploración de datos. Para preparar un conjunto de datos para entrenar un algoritmo de aprendizaje automático, es necesario comprender el conjunto de datos, limpiar y manipular los tipos y formatos de datos, completar los espacios vacíos en los datos y diseñar características para el algoritmo de aprendizaje. Estas tareas suelen agruparse bajo el término Extraer, Transformar, Cargar (ETL). El ETL suele ser un proceso iterativo y exploratorio. A medida que los conjuntos de datos crecen, la interactividad de este proceso se ve afectada cuando se ejecuta en CPU.

Para abordar los desafíos de la cadena de producción de ciencia de datos moderna, NVIDIA anunció hoy en GTC Europe RAPIDS , un conjunto de bibliotecas de software de código abierto para ejecutar cadenas de producción de ciencia de datos y análisis de extremo a extremo completamente en GPU. RAPIDS tiene como objetivo acelerar toda la cadena de producción de ciencia de datos, incluida la carga de datos, la extracción, transformación y carga (ETL), el entrenamiento de modelos y la inferencia. Esto permitirá flujos de trabajo más productivos, interactivos y exploratorios.

RAPIDS es el resultado de las contribuciones de la comunidad de aprendizaje automático y de los socios de la Iniciativa de Análisis Abierto de GPU (GOAI). Establecida en 2017 con el objetivo de acelerar los procesos de análisis de extremo a extremo y de ciencia de datos en las GPU, GOAI creó GPU DataFrame basándose en las estructuras de datos de Apache Arrow . GPU DataFrame permitió la integración de bibliotecas de aprendizaje automático y procesamiento de datos acelerados por GPU sin incurrir en las típicas penalizaciones de serialización y deserialización. RAPIDS se basa en el trabajo anterior de GOAI y lo amplía.

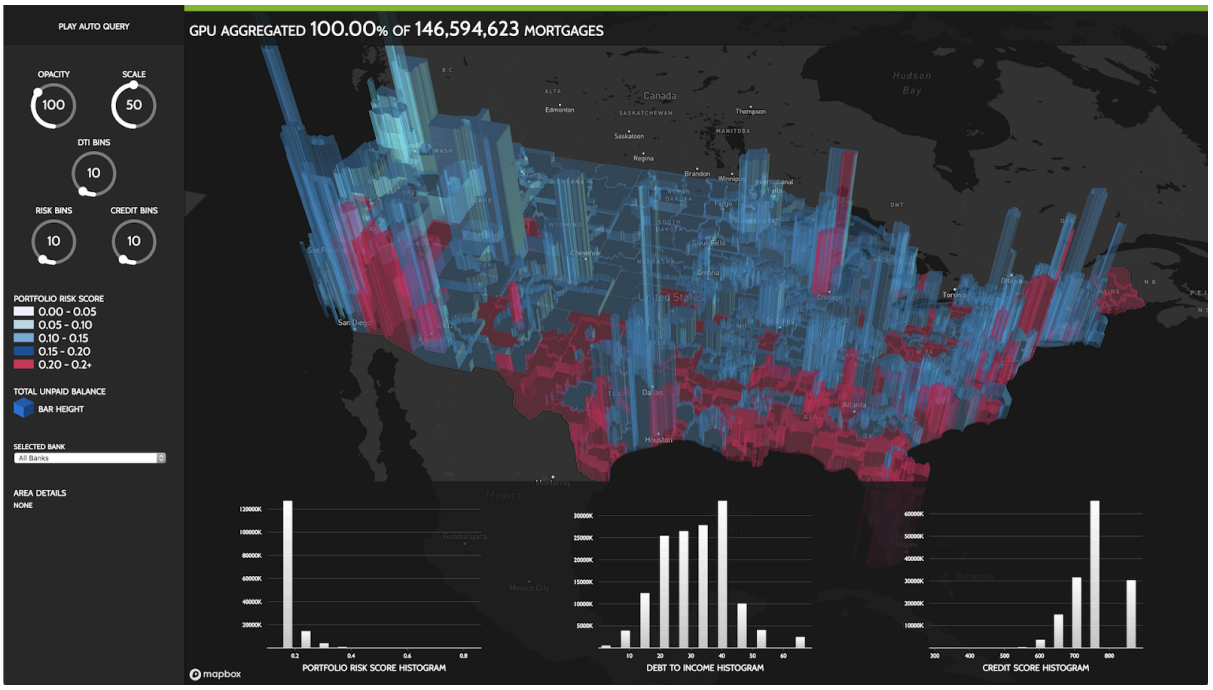
Mejore el rendimiento de la ciencia de datos con RAPIDS

RAPIDS logra factores de aceleración de 50x o más en flujos de trabajo típicos de ciencia de datos de extremo a extremo. RAPIDS utiliza NVIDIA CUDA para la ejecución de GPU de alto rendimiento, exponiendo ese paralelismo de GPU y el alto ancho de banda de memoria a través de interfaces Python fáciles de usar. RAPIDS se centra en tareas comunes de preparación de datos para análisis y ciencia de datos, y ofrece una API DataFrame potente y familiar. Esta API se integra con una variedad de algoritmos de aprendizaje automático sin pagar los costos de serialización típicos, lo que permite la aceleración de los procesos de extremo a extremo. RAPIDS también incluye soporte para implementaciones de múltiples nodos y múltiples GPU, lo que permite escalar hacia arriba y hacia abajo en tamaños de conjuntos de datos mucho más grandes.

El contenedor RAPIDS incluye un cuaderno y un código que demuestra un flujo de trabajo típico de ETL y ML de extremo a extremo. El ejemplo entrena un modelo para

realizar una evaluación de riesgo de préstamos hipotecarios utilizando todos los datos de préstamos de los años 2000 a 2016 en el conjunto de datos de desempeño de préstamos de Fannie Mae , que consta de aproximadamente 400 GB de datos en la memoria. La Figura 2 muestra una visualización geográfica del análisis de riesgo de préstamos.

https://capitalmarkets.fanniemae.com/credit-risk-transfer/single-family-credit-risk-transfer/fannie-mae-single-family-loan-performance-data



El ejemplo carga los datos en la memoria de la GPU mediante el lector CSV de RAPIDS. La ETL en este ejemplo realiza una serie de operaciones, entre las que se incluyen la extracción de meses y años de los campos de fecha y hora, uniones de varias columnas entre DataFrames y agregaciones agrupadas para la ingeniería de características. Los datos de características resultantes se convierten y se utilizan para entrenar un modelo de árbol de decisiones potenciado por gradiente en la GPU mediante XGBoost.

Este flujo de trabajo se ejecuta de extremo a extremo en un único servidor NVIDIA DGX-2 con 16 GPU Tesla V100, 10 veces más rápido que 100 instancias AWS r4.2xLarge, como lo muestra el gráfico de la figura 3. Si comparamos el rendimiento de la GPU con el de la CPU uno a uno, esto equivale a una aceleración de más de 50 veces.

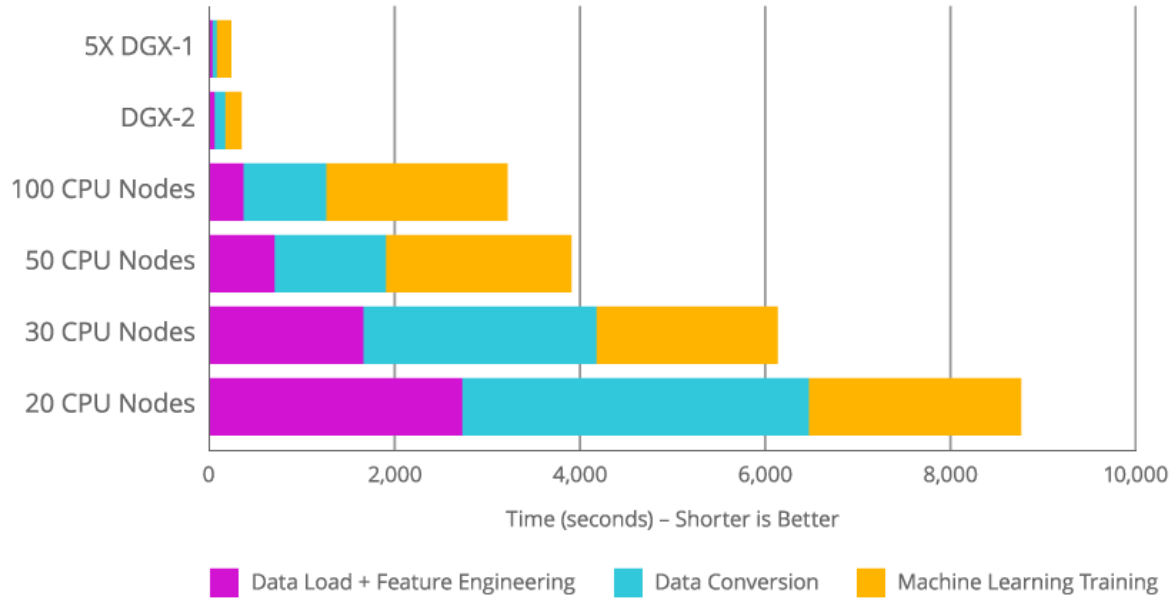


Figura 3. RAPIDS en NVIDIA DGX-2 proporciona aceleraciones de más de 50 veces en comparación con los clústeres de CPU en un flujo de trabajo de ciencia de datos típico.

La necesidad de una aceleración de extremo a extremo

La aceleración por GPU de algoritmos clásicos de aprendizaje automático, como el aumento de gradiente , ya se ha vuelto popular. Sin embargo, los esfuerzos anteriores para acelerar por GPU los procesos de ciencia de datos se han centrado en bibliotecas de aprendizaje automático individuales y no en otras piezas de interconexión cruciales del proceso. Esto crea un problema. Supongamos que su proceso tiene tres pasos:

https://xgboost.readthedocs.io/en/latest/gpu/index.html

1. Cargar datos
2. Limpiar los datos y realizar ingeniería de características
3. Entrenar un clasificador

Primero, se cargan los datos en la memoria del host. Luego, se realizan tareas de ETL, que incluyen la limpieza de datos y los pasos de ingeniería de características, como el filtrado, la imputación y la generación de nuevas características. En la actualidad, estos pasos se realizan en gran medida mediante la CPU. Después, se debe convertir la salida del paso de ingeniería de características al formato de memoria interna de la biblioteca de aprendizaje automático acelerada por GPU y, luego, mover los datos a la memoria de la GPU. Ahora, se ejecuta el entrenamiento. Se obtiene una gran aceleración en el paso de entrenamiento y se está satisfecho. Por último, se vuelven a mover los datos a la memoria del host y se visualizan o se preparan para la implementación.

Al final, se obtiene una modesta aceleración general, pero las operaciones de copia y conversión introducen una sobrecarga significativa debido a las operaciones de serialización y deserialización y se termina subutilizando la potencia de procesamiento de la GPU disponible.

RAPIDS resuelve este problema mediante su diseño. Proporciona una estructura de datos en columnas denominada GPU DataFrame, que implementa el formato de datos en columnas Apache Arrow en la GPU. El GPU DataFrame de RAPIDS proporciona una API similar a la de Pandas que resultará familiar para los científicos de datos, de modo que ahora pueden crear flujos de trabajo acelerados por GPU con mayor facilidad.

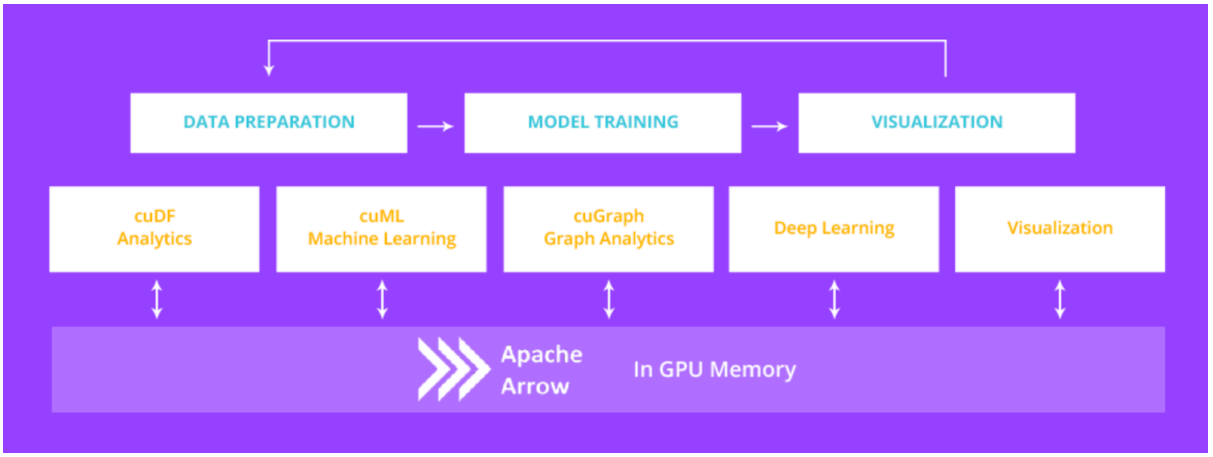
Bibliotecas de software RAPIDS

Los científicos de datos que utilizan RAPIDS interactúan con él utilizando los siguientes paquetes de Python de alto nivel.

- cuDF **:** una biblioteca GPU DataFrame con una API similar a la de Pandas. cuDF proporciona operaciones en columnas de datos, incluidas operaciones unarias y binarias, filtros, uniones y agrupaciones. cuDF actualmente comprende la biblioteca Python PyGDF y la implementación de GPU DataFrames de C++/CUDA en libgdf. Estas dos bibliotecas se están fusionando en cuDF. Consulte la documentación para obtener más detalles y ejemplos.
- cuSKL **:** una colección de algoritmos de aprendizaje automático que operan en marcos de datos de GPU. cuSKL permite a los científicos de datos, investigadores e ingenieros de software ejecutar tareas de aprendizaje automático tradicionales en GPU sin entrar en los detalles de la programación CUDA desde Python.
- XGBoost **:** XGBoost es uno de los paquetes de aprendizaje automático más populares para entrenar árboles de decisión potenciados por gradientes. La compatibilidad nativa con cuDF le permite pasar datos directamente a XGBoost mientras permanecen en la memoria de la GPU.

cuSKL es una biblioteca de cuML que permite que las siguientes bibliotecas de nivel inferior sean más accesibles para los desarrolladores de Python. Hay más información sobre estas bibliotecas disponible en la documentación. La Figura 4 destaca la estructura general y el flujo del pipeline utilizando RAPIDS.

- **cuML** : una biblioteca acelerada por GPU de algoritmos de aprendizaje automático que incluye descomposición en valores singulares (SVD), análisis de componentes principales (PCA) y agrupamiento espacial basado en densidad de aplicaciones con ruido (DBSCAN).
- **ml-prims** : una biblioteca de primitivas matemáticas y computacionales de bajo nivel utilizadas por cuML.



Obteniendo RAPIDS

El código fuente de RAPIDS está disponible en GitHub y hay un contenedor disponible en NVIDIA GPU Cloud (NGC) y Docker Hub . Veamos rápidamente cómo obtener el contenedor y ejecutarlo, y cómo acceder al cuaderno de flujo de trabajo de análisis de riesgo hipotecario.

Hay una imagen de contenedor Docker completa y lista para usar disponible en el registro de contenedores de Docker Hub de RAPIDS , lo que facilita comenzar a usar RAPIDS. Obtenga la última imagen de contenedor de RAPIDS ejecutando el siguiente comando:

```
$ docker pull nvcr . io / nvidia / rapidsai / rapidsai : último
```

Puedes verificar que tienes la imagen con el comando docker images:

```
$ docker images | grep rapids
rapids/rapidsai latest   4b30dcd9849c hace 2 días 8,51 GB
```

Ejecutar el contenedor RAPIDS

El contenedor puede iniciar automáticamente un cuaderno Jupyter o puede iniciar el contenedor en modo terminal agregando `bash`al final del comando Docker. Iniciemos el cuaderno.

```
$ docker run --runtime=nvidia \
        --rm -it \
        -p 8888:8888 \
        -p 8787:8787 \
        -p 8786:8786 \
        nvcr.io/nvidia/rapidsai/rapidsai:latest
```

Verás un `bash`indicador de terminal, donde podrás activar el `conda`entorno con el siguiente comando:

```
root @ 003 8283a49ef : / # cd rapids && source activar gdf
( gdf ) root @ 003 8283a49ef : / rapids #
```

Después de ejecutarlo, observe que se ha `(gdf)`añadido un mensaje de aviso para indicar el `conda`entorno activado. A continuación, debe descomprimir los datos proporcionados con <code>tar -xzvf data/mortgat/tar.gz, lo que da como resultado lo siguiente:

```
( gdf ) root @ 003 8283a49ef : / rapids # tar -xzvf data/hipoteca.tar.gz
hipoteca /
hipoteca / acq /
hipoteca / acq / Adquisición_2000T1 . txt
hipoteca / acq / Adquisición_2001T4 . txt
hipoteca / acq / Adquisición_2001T2 . txt
hipoteca / acq / Adquisición_2000T4 . txt
hipoteca / acq / Adquisición_2000T3 . txt
hipoteca / acq / Adquisición_2000T2 . txt
hipoteca / acq / Adquisición_2001T1 . txt
hipoteca / acq / Adquisición_2001T3 . txt
hipoteca / perf /
hipoteca / perf / Rendimiento_2001T2 . txt_0
hipoteca / perf / Rendimiento_2001T4 . txt_0
hipoteca / rendimiento / Rendimiento_2001T4 . txt_1
hipoteca / rendimiento / Rendimiento_2001T3 . txt_1”
hipoteca / rendimiento / Rendimiento_2000T1 . txt
hipoteca / rendimiento / Rendimiento_2001T1 . txt
```

hipoteca / rendimiento / Rendimiento_2000T4 . txt
hipoteca / rendimiento / Rendimiento_2000T3 . txt
hipoteca / rendimiento / Rendimiento_2000T2 . txt
hipoteca / rendimiento / Rendimiento_2001T3 . txt_0
hipoteca / rendimiento / Rendimiento_2001T2 . txt_1
hipoteca / nombres . csv

Ahora puede iniciar el servidor de cuadernos Jupyter, al que puede acceder desde la siguiente URL en su navegador: `{IPADDR}:8888 (e.g.) 12.34.567.89:8888`, donde `IPADDR`es la dirección de la máquina que ejecuta Docker. Donde dice "contraseña o token", ingrese "rapids". Encontrará un par de cuadernos de ejemplo en la `notebook`carpeta que ejecutarán ETL de extremo a extremo y aprendizaje automático en los datos proporcionados. La Figura 5 muestra el cuaderno ETL en ejecución. Desde aquí puede ejecutar y editar las celdas en el cuaderno (Shift+Enter) y ver el resultado de su ejecución.

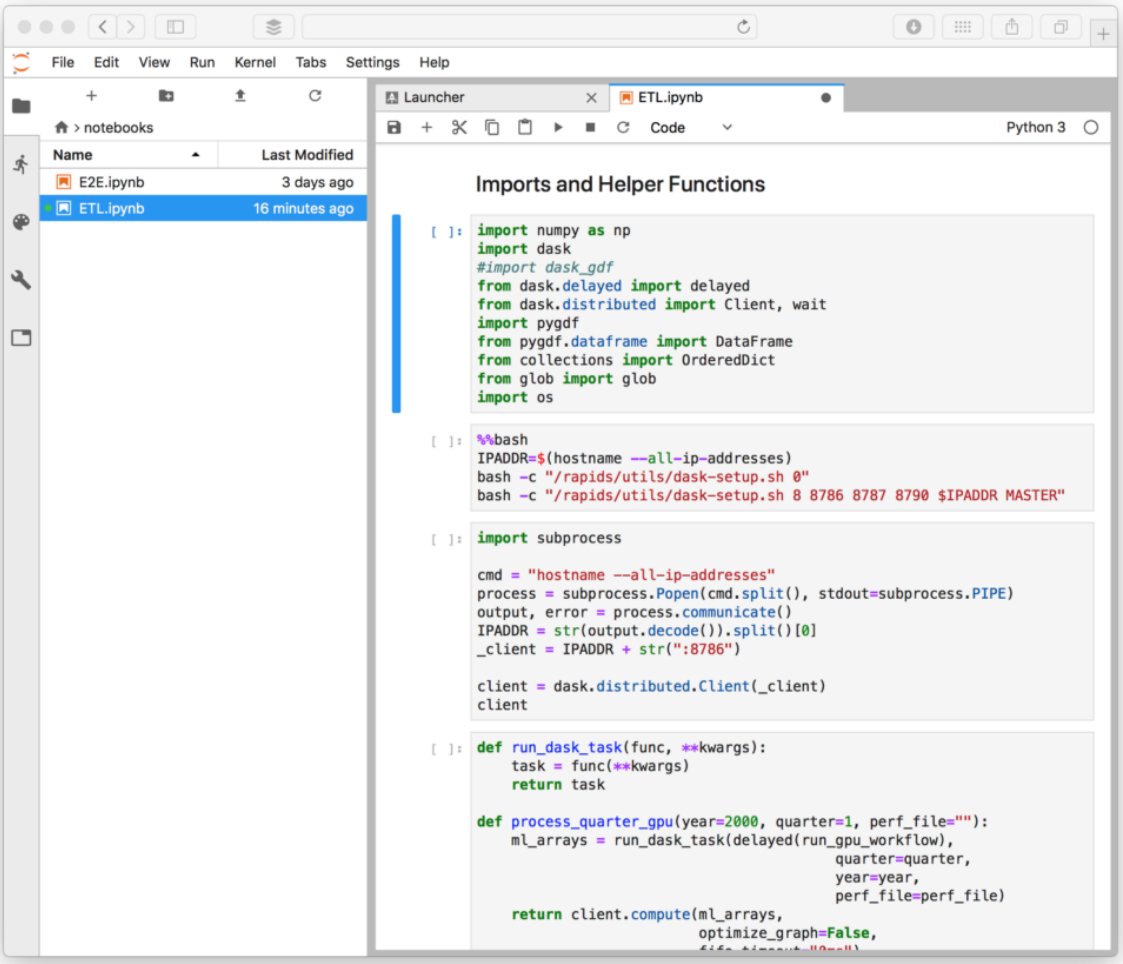


Figura 5: El contenedor Docker de RAPIDS contiene cuadernos de Python de ejemplo, incluido este cuaderno ETL de riesgo hipotecario que se ejecuta en múltiples GPU mediante Dask.

Revisaremos los detalles del ejemplo en detalle en una publicación futura para demostrar las API de Python y las capacidades de RAPIDS.

Tenga en cuenta que puede modificar lo anterior para experimentar con sus propios datos. Por ejemplo, puede iniciar una sesión interactiva con sus propios datos asignados al contenedor:

```
docker run - - runtime = nvidia \
    - - rm - it \
    - p 8888 : 8888 \
    - p 8787 : 8787 \
    - v / ruta / al / host / datos : / rapids / my_data
   nvcr . io / nvidia / rapidsai / rapidsai : latest
```

Esto asignará datos de su sistema operativo host al sistema operativo del contenedor en el directorio /rapids/my_data. Es posible que deba modificar los cuadernos proporcionados para las nuevas rutas de datos.
Puede obtener documentación interactiva sobre las funciones de Python en el cuaderno usando el `?`prefijo en el nombre de la función, como `?pygdf.read_csv`. Esto imprimirá la cadena de documentación para `read_csv`la función de PyGDF. Consulte la documentación de RAPIDS para obtener información más detallada y consulte el registro de contenedores de NVIDIA GPU Cloud para obtener más instrucciones sobre el uso del contenedor.

Conclusión

RAPIDS acelera todo el proceso de ciencia de datos, desde la ingestión y manipulación de datos hasta el entrenamiento de aprendizaje automático. Para ello, realiza lo siguiente:

1. Adopción de GPU DataFrame como formato de datos común en todas las bibliotecas aceleradas por GPU
2. Acelerar los componentes básicos de la ciencia de datos, como las rutinas de manipulación de datos que ofrece pandas y los algoritmos de aprendizaje automático como XGboost, mediante el procesamiento de datos y la retención de los resultados en la memoria de la GPU.

RAPIDS ahora está disponible como una imagen de contenedor en NVIDIA GPU Cloud (NGC) y Docker Hub para su uso en instalaciones locales o en servicios de nube pública como AWS, Azure y GCP. El código fuente de RAPIDS también está disponible en GitHub. Visite el sitio de RAPIDS para obtener más información. Si tiene preguntas, comentarios o sugerencias, utilice la sección de comentarios a continuación.

Recursos relacionados

Recursos relacionados

- **Curso DLI:** Aceleración de los flujos de trabajo de ciencia de datos de extremo a extremo
- **Sesión de GTC:** Aceleración de la ciencia de datos en Python con RAPIDS (primavera de 2023)
- **Kit de desarrollo de software:** Nsight Compute
- **Seminario web:** RAPIDS: ciencia de datos de última generación con NVIDIA
- **Seminario web:** Software informático acelerado NVIDIA HPC para mercados de capitales
- **Seminario web:** Cómo acelerar los flujos de trabajo de ciencia de datos con RAPIDS

https://docs.nvidia.com/ngc/ngc-titan-setup-guide/index.html#installing-docker-nv-docker

https://forums.developer.nvidia.com/t/rapids-accelerates-data-science-end-to-end/148622

https://forums.developer.nvidia.com/t/rapids-accelerates-data-science-end-to-end/148622

https://medium.com/rapids-ai
https://forums.developer.nvidia.com/t/rapids-accelerates-data-science-end-to-end/148622